# **SQL a Aggregation**

El pipeline de agregación permite a MongoDB proporcionar capacidades de agregación nativas que corresponden a muchas operaciones comunes de agregación de datos en SQL.

La tabla siguiente proporciona una descripción general de términos, funciones y conceptos comunes de agregación de SQL y los operadores de agregación MongoDB correspondientes:

|  |  |
| --- | --- |
| **SQL Terms, Functions, and Concepts** | **MongoDB Aggregation Operators** |
| WHERE | [$match](https://docs.mongodb.com/v3.2/reference/operator/aggregation/match/#pipe._S_match) |
| GROUP BY | [$group](https://docs.mongodb.com/v3.2/reference/operator/aggregation/group/#pipe._S_group) |
| HAVING | [$match](https://docs.mongodb.com/v3.2/reference/operator/aggregation/match/#pipe._S_match) |
| SELECT | [$project](https://docs.mongodb.com/v3.2/reference/operator/aggregation/project/#pipe._S_project) |
| ORDER BY | [$sort](https://docs.mongodb.com/v3.2/reference/operator/aggregation/sort/#pipe._S_sort) |
| LIMIT | [$limit](https://docs.mongodb.com/v3.2/reference/operator/aggregation/limit/#pipe._S_limit) |
| SUM() | [$sum](https://docs.mongodb.com/v3.2/reference/operator/aggregation/sum/#grp._S_sum) |
| COUNT() | [$sum](https://docs.mongodb.com/v3.2/reference/operator/aggregation/sum/#grp._S_sum) |
| join | [$lookup](https://docs.mongodb.com/v3.2/reference/operator/aggregation/lookup/#pipe._S_lookup)  *New in version 3.2.* |

## **Ejemplos**

La tabla siguiente presenta una referencia rápida de sentencias de agregación SQL y las instrucciones MongoDB correspondientes. Los ejemplos de la tabla asumen las siguientes condiciones:

* Los ejemplos SQL asumen *dos* tablas, orders y order\_lineitem que se unen por las columnas order\_lineitem.order\_id y orders.i.
* Los ejemplos de MongoDB asumen una orden de colección que contiene documentos del siguiente prototipo:
* {  
   cust\_id: "abc123",  
   ord\_date: ISODate("2012-11-02T17:04:11.102Z"),  
   status: 'A',  
   price: 50,  
   items: [ { sku: "xxx", qty: 25, price: 1 },  
   { sku: "yyy", qty: 25, price: 1 } ]  
  }

|  |  |  |
| --- | --- | --- |
| **SQL** | **MongoDB** | **Descripcion** |
| **SELECT** **COUNT**(\*) **AS** **count** **FROM** orders | db.orders.aggregate( [  {  $group: {  \_id: **null**,  count: { $sum: 1 }  }  } ] ) | Cuenta todos los registros de orders |
| **SELECT** **SUM**(price) **AS** total **FROM** orders | db.orders.aggregate( [  {  $group: {  \_id: **null**,  total: { $sum: "$price" }  }  } ] ) | Suma el campo price de orders |
| **SELECT** cust\_id,  **SUM**(price) **AS** total **FROM** orders **GROUP** **BY** cust\_id | db.orders.aggregate( [  {  $group: {  \_id: "$cust\_id",  total: { $sum: "$price" }  }  } ] ) | Para cada cust\_id unico, suma el campo price. |
| **SELECT** cust\_id,  **SUM**(price) **AS** total **FROM** orders **GROUP** **BY** cust\_id **ORDER** **BY** total | db.orders.aggregate( [  {  $group: {  \_id: "$cust\_id",  total: { $sum: "$price" }  }  },  { $sort: { total: 1 } } ] ) | Para cada cust\_id, suma el campo pric, los ordena por la suma. |
| **SELECT** cust\_id,  ord\_date,  **SUM**(price) **AS** total **FROM** orders **GROUP** **BY** cust\_id,  ord\_date | db.orders.aggregate( [  {  $group: {  \_id: {  cust\_id: "$cust\_id",  ord\_date: {  month: { $month: "$ord\_date" },  day: { $dayOfMonth: "$ord\_date" },  year: { $year: "$ord\_date"}  }  },  total: { $sum: "$price" }  }  } ] ) | Para cada cust\_id,ord\_date, suma el campo price. Excluye la parte de tiempo de la fecha. |
| **SELECT** cust\_id,  **count**(\*) **FROM** orders **GROUP** **BY** cust\_id **HAVING** **count**(\*) > 1 | db.orders.aggregate( [  {  $group: {  \_id: "$cust\_id",  count: { $sum: 1 }  }  },  { $match: { count: { $gt: 1 } } } ] ) | Para cust\_id, retorna el cust\_id y la cantidad de registros mayor a 1. |
| **SELECT** cust\_id,  ord\_date,  **SUM**(price) **AS** total **FROM** orders **GROUP** **BY** cust\_id,  ord\_date **HAVING** total > 250 | db.orders.aggregate( [  {  $group: {  \_id: {  cust\_id: "$cust\_id",  ord\_date: {  month: { $month: "$ord\_date" },  day: { $dayOfMonth: "$ord\_date" },  year: { $year: "$ord\_date"}  }  },  total: { $sum: "$price" }  }  },  { $match: { total: { $gt: 250 } } } ] ) | Para cada cust\_id,ord\_date los agrupa , sumando el campo price y solo retorna cuando la suma es mayor a 250. Excluye la parte del tiempo de la fecha |
| **SELECT** cust\_id,  **SUM**(price) **as** total **FROM** orders **WHERE** status = 'A' **GROUP** **BY** cust\_id | db.orders.aggregate( [  { $match: { status: 'A' } },  {  $group: {  \_id: "$cust\_id",  total: { $sum: "$price" }  }  } ] ) | Para cada unico cust\_id con status A, suma el campo price. |
| **SELECT** cust\_id,  **SUM**(price) **as** total **FROM** orders **WHERE** status = 'A' **GROUP** **BY** cust\_id **HAVING** total > 250 | db.orders.aggregate( [  { $match: { status: 'A' } },  {  $group: {  \_id: "$cust\_id",  total: { $sum: "$price" }  }  },  { $match: { total: { $gt: 250 } } } ] ) | Para cada unico cust\_id con status A, suma el price y retorna los valores mayores a 250. |
| **SELECT** cust\_id,  **SUM**(li.qty) **as** qty **FROM** orders o,  order\_lineitem li **WHERE** li.order\_id = o.id **GROUP** **BY** cust\_id | db.orders.aggregate( [  { $unwind: "$items" },  {  $group: {  \_id: "$cust\_id",  qty: { $sum: "$items.qty" }  }  } ] ) | Para cada unico cust\_id, suma los campos qty asociados conassociated with the orders. |
| **SELECT** **COUNT**(\*) **FROM** (**SELECT** cust\_id,  ord\_date  **FROM** orders  **GROUP** **BY** cust\_id,  ord\_date)  **as** DerivedTable | db.orders.aggregate( [  {  $group: {  \_id: {  cust\_id: "$cust\_id",  ord\_date: {  month: { $month: "$ord\_date" },  day: { $dayOfMonth: "$ord\_date" },  year: { $year: "$ord\_date"}  }  }  }  },  {  $group: {  \_id: **null**,  count: { $sum: 1 }  }  } ] ) |  |